Amendments to the Specification

Please replace the abstract with the following amended abstract:

The present invention provides a system and method for encapsulating protocols across a switching fabric network. Packets, which may utilize any underlying protocol, are encapsulated with a route header. This route header contains path routing, traffic and packet size information. In one embodiment, a novel path routing scheme is used to route packets across the fabric, where the fabric has a plurality of switches, each having a plurality of ports. Each switch uses only data from within the packet and its own port count to determine the appropriate output port. There is no need to node or address lookup mechanisms in the switches. In another embodiment, a multicast routing system is disclosed. Mechanisms which allow for detection of routing errors are also disclosed.

Please replace paragraph [0020] with the following amended paragraph:

[0020] A turn is a variable sized, contiguous field containing one or more bits. The actual value of a turn is called its turn value. A turn value indicates the position of an output port relative to the input port on which a packet is received, and contains the number of relative ports to be skipped between the input port and the output port. An output port number is equal to (*[input_port_number+turn_value+1] modulo* ~~*[N^2+1]*~~ *[2^N+1]*). For example, the turn value for a packet entering a switch on port x and leaving on port x+1 is 0, and the turn value for a packet entering port x and leaving port x+4 is 3.

Please replace paragraph [0021] with the following amended paragraph:

[0021] The switches in an ExAS fabric can be classified by the number of bits, of turn pool they consumed. Logically each switch has ~~N^2+1~~ 2^N+1 ports, where N is the number of bits of turn pool consumed by the switch. For example, a 3-bit switch logically has 9 ports, even if physically it only had 7 ports. From a relative routing perspective, port 0 follows port ~~N^2~~ 2^N, or a turn value of 0 in a packet that arrived on physical port ~~N^2~~ 2^N would indicate that the packet should exit the switch on physical port 0. A switch cannot use more physical ports than its turn pool consumption indicates. For example a 4-bit switch can have at most 17 physical ports. Conversely, a switch could consume 5-bits of turn pool but only contain say 6 physical ports, though profligate consumption of turn pool bits is strongly discouraged. The sizes of the turns within a turn pool is determined by the switches that process the turn pool as it traverses a fabric. FIG. 3 shows the routing portion of a unicast header.

Please replace paragraph [0037] with the following amended paragraph:

[0037] Multicast packets are routed from node to node based on table look-ups at each switch. Each multicast packet has a Multicast Group Index, which identifies ~~to~~ the multicast group to which the packet belongs. This 12-bit Multicast ID occupies bits 63-48 of the ExAS routing header. The bit count field of the unicast header is recast as a hop count field for multicast headers. A multicast distribution tree is limited to 22 switch traversals (hops). When a multicast packet is received at a switch, its hop count field (bits 24-19 of the ExAS routing header) is tested. If the hop count is 22, then a multicast routing error has occurred. If the hop count is 21 or less, then the hop count is incremented and the packet is forwarded.

Please replace paragraph [0048] with the following amended paragraph:

[0048] Note that a single multicast packet can result in the generation of many packets within the fabric. A switch may generate up to ~~N^2~~ 2^N multicast packets (where N is the bit number of the switch) for each multicast packet it receives. Applications need to be aware of this characteristic of multicast traffic and plan bandwidth usage accordingly.

Please replace paragraph [0065] with the following amended paragraph:

[0065] FIG. 11 ~~show and~~ shows an example of a simple read data packet format with ExAS Routing Header Attached shown for purposes of clarity. FIG. 12 shows a simple read data packet format. The simple read data packet format is further described in Table 8.